ITRI 613 - Assignment 2
Due Date: 13/04/2021 (MEMO)

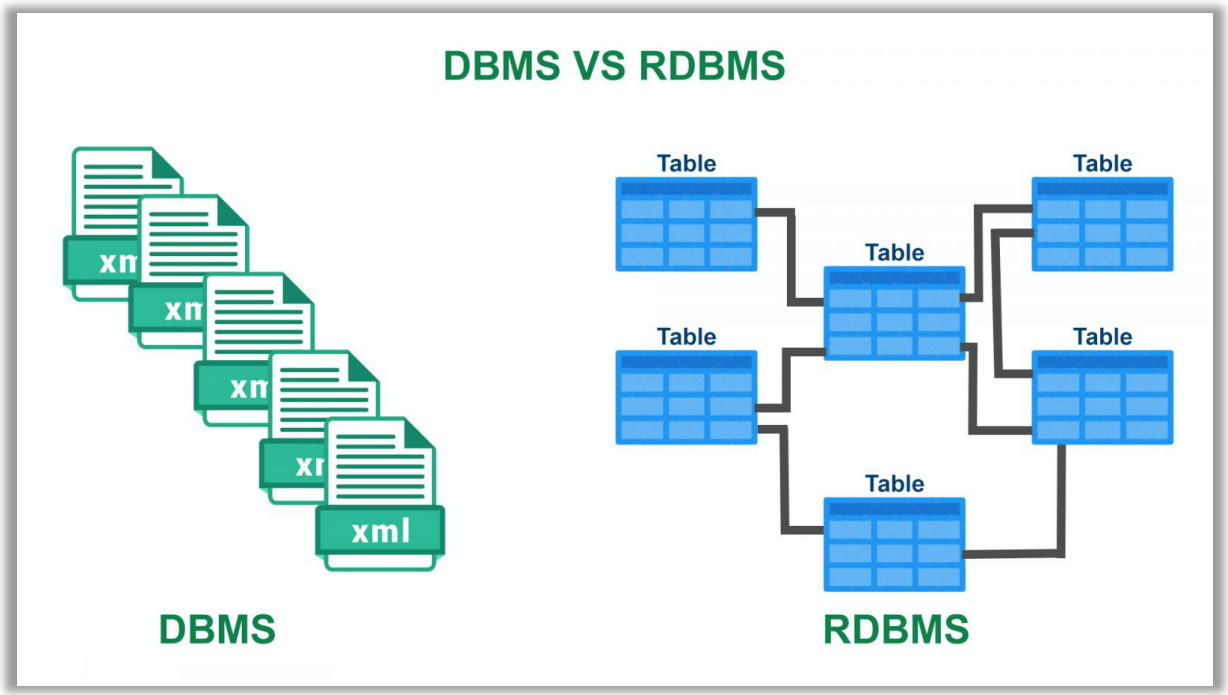
RDBMS

Relational Database Management System (RDBMS) is an advanced version of a DBMS system. It came into existence during 1970's. RDBMS system also allows the organization to access data more efficiently than DBMS. One of the most important features of a RDBMS is the ability to support multiple users whereas the DBMS only supports one user at a time.

Most companies today have switched from using DBMS to use RDBMS because of its advanced capabilities and its abilities to help business handle data and manage information by storing it in the form tables.

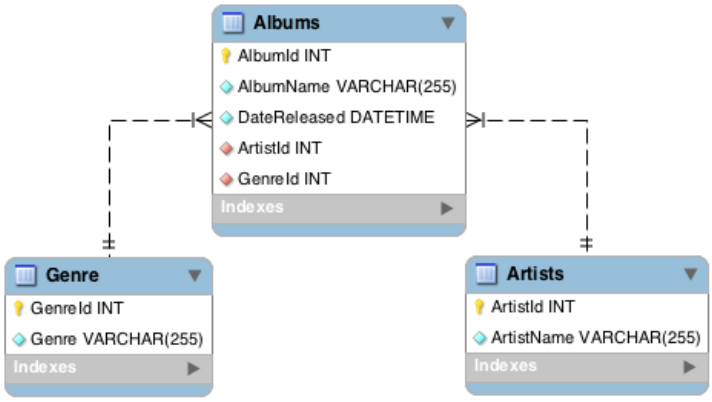
PART 1

For the first part of the assignment, Outline the any five RDBMS packages of your choosing and discuss their unique features and characteristics and also explains how they are better than traditional DBMS packages.



PART 2

Given the **MusicDB** Schema below for a popular music streaming application. Refer to the schema and answer the following questions.



1. Write an SQL query to display all attributes from Genre table.
   Select * from genre;

2. Write an SQL query to drop the table Artists from the DB.
   Drop table Artist;

3. Write an SQL query to create the table AlbumSales which will have attributes (ArtistId INT, AlbumId INT, NumberofSales INT, Genre VARCHAR).

   CREATE TABLE AlbumSales (
   ArtistId INT,
   AlbumId INT,
   NumberofSales INT,
   Genre VARCHAR (255),
   PRIMARY KEY (ArtistId,AlbumId),
   FORGEIN KEY (ArtistId) REFERENCES Artists (ArtistId),
   FORGEIN KEY(AlbumId) REFERENCES Albums (AlbumId);

4. Write an SQL query to return all albums which were released in year 2020 from the table Albums (i.e Where attribute DateReleased is of year 2020).

   SELECT * FROM Albums WHERE YEAR(DateReleased) = '2020';

5. Write an SQL query to create the table **Albums** but exclude the attribute **GenreId** and only have **ArtistId** as a primary key. Also explains what will happen to the Table **Genre** if this is the case.

```
CREATE TABLE Albums(
AlbumId INT PRIMARY KEY NOT NULL,
AlbumName VARCHAR(255) NOT NULL,
DateReleased DATETIME NOT NULL,
ArtistId INT NOT NULL,
FOREIGN KEY (ArtistId) REFERENCES Artists (ArtistId)
);
```

6. Using the above schema provide an example scenario in which an **overlapping constraint** may be experienced. (Hint – Create additional tables named **Singles, ExtendedPlay**)

   The simple example to explain this is that an overlapping constraint may be experienced because an artist can have both singles and an extended play released at any given instant and have songs from ExtendedPlay entity as singles.

7. Write SQL query for the whole schema and include the tables **AlbumSales** created above in question 3. Also include an additional entities named **ExtendedPlay** which has attributes **(ArtistId INT, EPId INT, DateReleased DATETIME, Genre VARCHAR).**

```
CREATE TABLE Genre (
GenreId INT PRIMARY KEY NOT NULL,
Genre VARCHAR(255) NOT NULL,
);

CREATE TABLE Artists (
ArtistId INT PRIMARY KEY NOT NULL,
ArtistName VARCHAR(255) NOT NULL
);

CREATE TABLE Albums (
AlbumId INT PRIMARY KEY NOT NULL,
AlbumName VARCHAR(255) NOT NULL,
DateReleased DATETIME NOT NULL,
ArtistId INT NOT NULL,
GenreId INT NOT NULL,
```

```
FOREIGN KEY (ArtistId) REFERENCES Artists (ArtistId),
FOREIGN KEY (GenreId) REFERENCES Genre (GenreId)
);

CREATE TABLE AlbumSales (
ArtistId INT PPRIMARY KEY NOT NULL,
AlbumId INT NOT NULL,
NumberofSales INT,
Genre VARCHAR (20),
FOREIGN KEY (ArtistId) REFERENCES Artists (ArtistId),
FOREIGN KEY (AlbumId) REFERENCES Albums (AlbumId)
);

CREATE TABLE ExtendedPlay (
EPId INT PPRIMARY KEY NOT NULL,
ArtistId INT NOT NULL,
DateReleased DATETIME,
Genre VARCHAR(20),
FOREIGN KEY (ArtistId) REFERENCES Artists (ArtistId),
);
```

8. Suppose there was an additional entity named **RecordSales** which had attributes such as (**ArtistId INT, AlbumId INT, NumberofSales INT, Genre VARCHAR, GoldStatus VARCHAR, PlatinumStatus VARCHAR**) what kind of relationship type will it have with the table **Albums**?
   One to one relationship.

9. Create a View which displays all attributes from the table **Artists**?

```
CREATE VIEW [Artist View]
SELECT * FROM Artist;
```

10. List all the **1 to 1** and **1 to many** relationships from the schema with all the additional tables added (i.e **RecordSales, ExtendedPlay, Singles**).

    One-to-one - RecordSales and Albums, RecordSales and Artists
    One-to-many - Albums and Genre, Albums and Artists, Artists and single, Artists and ExtendedPlay